# 海康威视 DS-400xH 系列卡系统 SDK 说明书

**（FOR LINUX）**

(2004-11-12 3.8 版)

海康威视 DS-400xH 是面向数字监控行业而推出的专用板卡，采用了高性能的视频压缩技术标准 H.264 及 G.722 的音频编码标准，完全依靠硬件实现了视频及音频的实时编码（CIF 格式 25 帧）并精确同步，实现了动态码率、可控帧率、帧模式选择、动态图像质量控制，音频预览、视频丢失报警、能独立调整各通道参数，性能稳定而且可靠。与 MPEG-I 产品相比，在保持同等图像质量的前提下，能大大节省存储空间、并非常适合宽带网或窄带网的传输，是新一代数字监控产品的最佳选择。

海康威视 DS-400xH 系列卡 SDK 分为三部分，分别为系统 SDK、网络 SDK、播放 SDK，本文档专门描述系统 SDK，其他 SDK 请参照我公司相关文档。系统 SDK 是专门为该系列一路及多路板卡设计的本地录像软件接口程序，以动态库的形式（libtmsdk.so）提供给应用软件开发者，并同时附有演示程序及其源码，能有效地缩短应用软件的开发周期。

在使用过程中，特别提醒软件开发人员，DS-400XH 系列压缩卡可动态设置帧结构、帧率和图像质量，即在压缩过程中可改变帧率（SetIBPMode(…)）和量化系数（SetDefaultQuant）无须停止、启动压缩，还是保持同一数据流记录。播放器会自动识别帧率等参数，按当前压缩帧率播放且声音播放保持正常。

通过动态修改量化系数（I、B、P）可控制压缩码率，当码率太高时，加大量化系数；当码率太低时，减少量化系数。当然，在量化系数满足的情况下，不必再降低量化系数。

DS-400XH 系列压缩卡的运动检测独立于压缩，不进行压缩也可以进行运动检测。可动态改变帧率非常有价值，在运动时按 25F/S 录像，在无运动时按较低帧率录像，运动时按高帧帧率录像，记录在同一个文件内，可大大节省硬盘空间。

SetLogo(…)不仅可作 LOGO 使用，还可以用于遮盖图像中的某些区域。

注意事项：**本系统 SDK 的使用环境为 Red Hat 7.2 Kernel 2.4.7-10**
**或 Red Hat 7.3 Kernel 2.4.18-3**
**或 Red Hat 8.0 Kernel 2.4.18-20**
**或 Red Hat 9.0 Kernel 2.4.20-8**
**或 Federa1 Kernel 2.4.22-1**

**如需运行我们提供的 DEMO 程序，需要 XWINDOW 支持。**

**安装完驱动后，必须重新启动计算机。**

## 版本升级说明：

## 3.8 版本改进部分:

1. 增加了新的编码分辨率：Double CIF(ENC_DCIF_FORMAT)。PAL：528*384，NTSC：528*320。DCIF 和CIF 相比，在相同的码流下，图像质量和线数会有明显提高；
2. 升级码流的格式，可以支持任意改变录像分辨率而不用切换文件(需要配合新的解码库使用)；
3. 新增了屏幕遮挡MASK 函数，最大支持32 个区域，相对坐标范围（0，0，703，

575）；
4.预览提供更多选择，提供了新接口StartVideoPreviewEx（），现可支持非固定1024*768 、16位色的分辨率，现可支持16位色与24位色。在RedHat9.0下特别提供了overlay 的方式：根据sdl的特性，现提供两种方式，软加速与硬加速的方式。在软加速时，操作相对简单，但cpu占用率高，在硬加速时，cpu占用率相对较低，但操作较复杂。 具体使用请参看分别的demo: softdemo.c 和 harddemo.c
5.在双编码模式时，CIF 与QCIF(QQCIF)的帧率分别可调，但QCIF(QQCIF)的帧率不能大于CIF 的； 在双编码模式时， CIF 为设为复合流时，QCIF 可为复合流或视频流，当CIF 为视频流，QCIF 只能设为视频流；

**3.4 版本改进部分：**

**供了双编码模式，即同时提供CIF＋QCIF/QQCIF数据流；**

**logo位图最大支持128*128；**

## 3．2 版本改进部分：

- 提供了 704*288 即 2CIF 的分辨率的编码模式(通过 SetEncodePictureFormat 设置)；
- 提供了 OSD 按照背景亮度自动调整显示亮度的功能；
- 音频和视频数据可以单独处理，可用于实现音频优先的传输方案；
- 提供了在 B 帧编码模式下的帧率控制；
- 改进音频的处理，保证音频的流畅；
- OSD 能显示 44 个 ASCII 字符；
- 本地支持 24 路音视频编码；

## SDK3.0 与以前版本，即与海康威视前期推出的 DS-400xM 相比，DS-400xH 系列主要特点有：

- 在保持相同图像质量的前提下，与 DS-400xM 系列板卡相比，压缩码流降低 30%以上，在办公室典型环境中 码率只有 20kbps~120kbps。
- 提供了非常精确的码率控制方式，无论在何种情况下均能输出指定的码率，并增加了 CBR 控制方式。
- 极大地降低了由于摄像噪声导致的图像失真、背景游动等现象。
- 采用了 G.722 的音频压缩算法，声音更加流畅。
- 将会提供高分辨率（640*480）的视频压缩功能。
- SDK 接口完全与 DS-400xM 系列板卡一致，并增加了一些其他功能。已开发的应用软件可以迅速完成移植。

SDK2.2,与 1.2 版相比的主要改进内容：

- 图像质量进一步提高；
- 修改 SetOsd()的接口定义，去掉了接口参数 VerticalOffset；
- 改进 OSD 的设置方式，提供了 OSD 位置、亮度、格式以及用户可编写的字符串；

- 改进了 LOGO 的设置方式，可以显示背景完全透明的或不透明的 BMP 图像；
- 增加了码率控制，可以限制编码的最高码率；
- QCIF 模式可以在录像停止后设置，不必重新启动系统
- 改进了 QCIF 编码的清晰度；
- 修改了一些 BUG；

SDK 版本 1.2，与 1.0 版相比的主要改进内容：

- 图像质量有一定的提高(提高了编码及回放图像的清晰度)；
- 修改了 SetOsd 函数的接口定义
- 增加了一些接口，具体请查看函数说明：
  1．CaptureIFrame 提供单 I 帧捕获，不会影响实时录象
  2．SetupMotionDetection 设置运动检测区域
  3．MotionAnalyzer 分析运动检测数据
  4．AdjustMotionDetectPrecision 运行时调整运动检测精度
  5．SetEncodePictureMode 设置图像编码尺寸 CIF 或 QCIF，QCIF 适合窄带传输如 PSTN
- 改进了运动侦测，运动分析灵敏可靠（注意：原运动向量接口关闭使用，改为运动强度以简化软件设计要求）；
- 系统目前支持最大 16 路的视音频录像；
- 修改了 1.0 版的一些 BUGs

## 1. 本 SDK 使用的错误代号定义及说明：

| 错误号 | 解释 |
|---|---|
| TMSDKErrorCodeDSPUninit | DSP 未初始化 |
| TMSDKErrorCodeDSPOpenFail | DSP 打开失败 |
| TMSDKErrorCodeDSPStopFail | DSP 停止失败 |
| TMSDKErrorCodeDSPLoadFail | DSP 程序下载失败 |
| TMSDKErrorCodeDSPStartFail | 启动 DSP 失败 |
| TMSDKErrorCodeDSPStopUnexpect | DSP 非正常关闭 |
| TMSDKErrorCodeDSPRestFail | DSP 复位失败 |
| TMSDKErrorCodeDSPCloseFail | DSP 关闭失败 |
| TMSDKErrorCodeDSPMagicMismatch | 非法的 DSP 句柄 |
| TMSDKErrorCodeDSPMessageCreateFail | 创建 MESSAGE 失败 |
| TMSDKErrorCodeDSPSyncObjectCreateFail | 创建 SYNCOBJECT 失败 |
| TMSDKErrorCodeDSPMessageSendFail | 发送 MESSAGE 失败 |
| TMSDKErrorCodeDSPMessageReceiveFail | 接收 MESSAGE 失败 |
| TMSDKErrorCodeDSPSharedMemoryCreateFail | 创建 SHAREMEM 失败 |
| | |
| TMSDKErrorCodeInvalidArgument | 非法参数 |
| TMSDKErrorCodeNoCardInstalled | 系统未安装 DS-400xM 系列卡 |
| TMSDKErrorCodeCreateCommInfoFail | 初始化系统参数失败 |

| | |
|---|---|
| TMSDKErrorCodeMemLocateFail | 分配内存失败 |
| TMSDKErrorCodeChannelNumOutOfRange | 非法的通道号 |
| TMSDKErrorCodeChannelMagicMismatch | 非法的通道句柄 |
| TMSDKErrorCodeUseChannelClosed | 试图使用已关闭的通道 |
| TMSDKErrorCodeGetVgaBaseFail | 无法得到 VGABASE |
| TMSDKErrorCodeFileOpenFail | 打开文件失败 |
| TMSDKErrorCodeCreateThreadFail | 建立线程失败 |
| TMSDKErrorCodeDriverNotInstalled | 系统未安装 DS-400xM 系列卡的驱动 |
| TMSDKErrorCodeCanNotDeInitDSPsWhileChannelOpen | 试图在仍有打开通道的情况下关闭 DSP |
| TMSDKErrorCodeTryToOpenChannelTwice | 试图两次打开同一通道 |
| TMSDKErrorCodeTryToCloseChannelInUse | 试图关闭正在使用(PREVIEW OR CAPTURE)的通道 |
| TMSDKErrorCodeWrongVerCore | DSP 程序的版本号与 SDK 的版本号不匹配 |
| TMSDKErrorCodeTimeOut | 超时错误 |

# 2. 数据类型定义

## 2.1 视频预览输出格式：

vdfRGB16 16 位 RGB 视频压缩格式
vdfRGB24 24 位 RGB 视频压缩格式
vdfYUV422Planar YUV422 视频压缩格式

## 2.2 帧类型定义

PktSysHeader 系统头
PktIFrames I 帧包
PktPFrames P 帧包
PktBBPFrames BBP 帧包
PktAudioFrames 音频帧包
PktMotionDetection 动态监测包
PktSFrames 1.2 版新增类型，为 I 帧捕获时传送的帧类型

## 2.3 视频标准定义

StandardNone 无视频信号
StandardNTSC NTSC 制式
StandardPAL PAL 制式

## 2.4 流类型

STREAM_TYPE_VIDEO 视频流
STREAM_TYPE_AUDIO 音频流
STREAM_TYPE_AVSYNC 音视频同步流

# 3. 数据结构定义

## 3.1 特殊功能能力定义

```
typedef struct tagChannelCapability{
      UCHAR bAudioPreview;              音频预览
      UCHAR bAlarmIO;                   报警信号
      UCHAR bWatchDog;                  看家狗
}CHANNEL_CAPABILITY, *PCHANNEL_CAPABILITY;
```

## 3.2 帧数据统计

```
typedef struct tagFramsStatistics{
      ULONG VideoFrames;                视频帧
      ULONG AudioFrames;                音频帧
      ULONG FramesLost;                 丢失帧
      ULONG QueueOverflow;              缓存溢出
}FRAMES_STATISTICS, *PFRAMES_STATISTICS;
```

## 3.3 显示区域

```
typedef struct tagRect{
        short RectTop;
        short RectBottom;
        short RectLeft;
        short RectRight;
}RECT;
```

## 3.3 版本信息

```
typedef struct tagVersion{
      ULONG DspVersion, DspBuildNum;              DSP 版本及 BUILD 号
      ULONG DriverVersion, DriverBuildNum;        驱动版本及 BUILD 号
      ULONG SDKVersion, SDKBuildNum;              SDK 版本及 BUILD 号
}VERSION_INFO, *PVERSION_INFO;
```

## 3.4 运动检测数据帧头结构

| | |
|---|---|
| MotionDataHeader 头 | 运动强度参数（短整型，长度为 BlockSize）见 3.5 节 |

```
typedef struct tagMotionData{
      PictureFormat_t PicFormat;                  图像类型：2CIF、CIF 或 QCIF
      ULONG HorizeBlocks;                         水平块个数（2CIF 为 44，
```

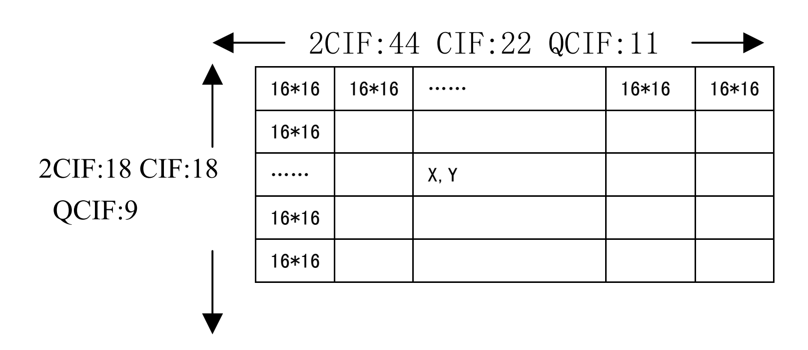
| | |
|---|---|
| | CIF 为 22，QCIF 为 11） |
| ULONG VerticalBlocks; | 垂直块个数（2CIF 为 18，CIF 为 18，QCIF 为 9） |
| ULONG BlockSize; | 总共块个数（2CIF 为 792，CIF 为 396，QCIF 为 99） |

}MOTION_DATA_HEADER, *PMOTION_DATA_HEADER

3.5 运动强度数据结构

为了简化运动检测的软件复杂度，DSP 软件现在提供运动强度信息来处理运运动检测，经过测试，这种方法比 1.0 版提供的运动向量更加灵敏和可靠，能可靠的处理在光照不足的情况下运动物体，数据结构定义如下：

```
unsigned short MotionData[VertivalBlocks][HorizBlocks]
```




运动强度定义

16x16 的宏块寻址方式为：

MotionData[y * HozrizBlocks + x];

取值范围：

0- 100

运动条件：

当宏块的数值 >= 0 时为该宏块发生运动，运动幅度为返回值的大小。

;

# 4. 函数说明

4.1 int InitDSPs();

说明：初始化每块板卡，应在应用软件启动时完成，如果返回值为-1 表明初始化失败，可能是没找到相应的 DSP 软件模块，其对应的接口为 DeInitDSPs；

返回：成功时返回 0，否则返回-1；

4.2 int DeInitDSPs();

说明：关闭每块板卡上功能，应在程序退出时调用；

返回：成功时返回 0，否则返回-1；

4.3 int ChannelOpen(int ChannelNum, void * StreamReadCallBack);

参数：1. int ChannelNum 通道号（从 0 开始）；

2.void * StreamReadCallBack 用户定义的流处理回调函数

（流处理函数定义的具体要求见 5.1）

说明：打开通道，注册流处理回调函数，获取相关的操作句柄，与通道相关的操作必须使用该句柄；

返回：成功时返回有效句柄，失败时返回 0;

4.4 int ChannelClose(int hChannelHandle)

参数：int hChannelHandle 通道句柄

说明：关闭通道，释放相关资源；

返回：成功时返回 0，否则返回-1；

4.5 int GetTotalChannels()

返回：获取系统内可使用的通道个数；

4.6 int GetTotalDSPs();

返回：获取系统内正确安装的 DSP 个数；

4.7 int StartVideoPreview(int hChannelHandle, RECT PreviewRect, int VideoFormat, int FrameRate);

参数： int hChannelHandle 通道句柄
RECT PreviewRet, 矩型显示区域
int VideoFormat, 视频格式（见 2.1 节）（暂无效）
int FrameRate 视频帧率（暂无效）

说明： 启动视频预览，视频格式定为 vdfRGB16；

返回： 成功时返回 0，否则返回-1；

4.8 int StopVideoPreview(int hChannelHandle)

参数： int hChannelInfo 通道句柄

说明： 停止视频预览；

返回： 成功时返回 0，否则返回-1；

4.9 int SetVideoPara(int hChannelHandle, int Brightness, int Contrast, int Saturation, int Hue)

参数： int hChannelHandle, 通道句柄
int Brightness, 亮度值（0—255）
int Contrast, 对比度（0—127）
int Saturation, 饱和度（0—127）
int Hue 色调 （0—255）

说明： 设置视频参数；

返回： 成功时返回 0，否则返回-1；

4.10 int GetVideoPara(int hChannelHandle, VideoStandard_t * VideoStandard, int *Brightness, int *Contrast, int *Saturation, int *Hue);

参数： int hChannelHandle, 窗口句柄
VideoStandard_t * VideoStandard 视频标准（见 2.3）

int *Brightness, 亮度指针值 （0—255）
int *Contrast, 对比度指针值（0—127）
int *Saturation, 饱和度指针值（0—127）
int *Hue 色调指针值 （0—255）
说明： 得到视频参数；
返回： 成功时返回 0，否则返回-1；

4.11 int GetSDKVersion()
参数：PVERSION_INFO VersionInfo 指向 VERSION_INFO 的参数指针
说明： 获取当前使用的 SDK 版本号；
返回： 高 16 位为主版本，低 16 位次版本号，

4.12 int GetCapability(int hChannelHandle, CHANNEL_CAPABILITY *Capability);
参数： int hChannelHandle, 通道句柄
CHANNEL_CAPABILITY *Capability 见 3.1 节
说明： 获取板卡的特殊功能信息；
返回： 成功时返回 0，否则返回-1;

4.13 int GetLastErrorNum();
说明： 获取最近一次的 SDK 及 DSP 错误号；
返回： 第 1 节定义的错误号;

4.14 int SetStreamType(int hChannelHandle, int StreamType);
参数： int hChannelHandle, 通道句柄
int StreamType 流类型，见以下宏定义
宏定义：
#define STREAM_TYPE_VIDEO 1 //视频流
#define STREAM_TYPE_AVSYNC 3 //音视频同步流
说明： 设置流类型；
返回： 成功时返回 0，否则返回-1；

4.15 GetStreamType(int hChannelHandle, int *StreamType);
参数： int hChannelHandle 通道句柄
int *StreamType 指向流类型指针
说明： 获得流类型；
返回：成功时返回 0，否则返回-1；

4.16 int GetFramesStatistics(HANDLE hChannelHandle, PFRAMES_STATISTICS framesStatistics);
参数： int hChannelHandle, 通道句柄
PFRAMES_STATISTICS framesStatistics 帧统计信息（见 3.2 节）
PFRAMES_STATISTICS 结构：
说明： 获取帧统计信息；
返回： 成功时返回 0，否则返回-1；

4.17 int SetupMotionDetetion(int hChannelHandle, RECT *rectList, int numberOfAreas)

参数：　int hChannelHandle,　　　　　　　　通道句柄

　　　　RECT *rectList,　　　　　　　　　　矩形框数组

　　　　int numberOfAreas,　　　　　　　　矩形框个数

说明：　设置运动检测区域，当收到运动信息的数据帧（PktMotionDetection）时,调用 MotionAnalyzer, MotionAnalyzer 会根据在 SetupMotionDetection 中的设置来分析每个需要检测的区域，当某区域的阀值(MotionAnalyer 的 iThreshold)到达时，会在返回的区域数组（MotionAnalyzer 的 iResult）标明最后的判断；

返回：　成功时返回 0，否则返回-1；

4.18　int StartMotionDetection(int hChannelHandle);

参数：　int　hChannelHandle　　　　　　　通道句柄

说明：　启动运动检测，运动检测信息会通过数据流传送，用户程序发现是 PktMotionDetection 帧类型时，可调用 MotionAnalyzer 来处理运动信息，结果由 MotionAnalyzer 在 iResult 中返回。也可以按照 SDK 提供的数据格式来自己分析，运动信息格式参见 3.4 和 3.5 节。注意：运动检测与编码相互独立，用户程序可在不启动编码的情况下进行运动检测。

返回：　成功时返回 0，否则返回-1；

## 注：该函数在 v3.0 版中有所改动

4.19 int GetBoardInfo(int hChannelHandle, int *BoardType, int *SerialNo);

参数：　int hChannelHandle,　　　　　　　通道句柄

　　　　int *BoardType,　　　　　　　　　板卡型号(输出参数)

　　　　int *SerialNo　　　　　　　　　　板卡 ID 号（输出参数）

在新版本中改为：

int GetBoardInfo(int hChannelHandle, int *BoardType, char *SerialNo);

参数：　int hChannelHandle,　　　　　　　通道句柄

　　　　int *BoardType,　　　　　　　　　板卡型号，DS-400xM/S 为 0，DS-400xH 为 1

　　　　char *SerialNo　　　　　　　　　板卡 ID 号，内容为板卡序列号的 ascii 的数值，次序为 SerialNo[0] 对应最高位，SerialNo[11]对应最低位。比如卡号为"4 0 0 0 0 1 0 0 2 3 4 5"的值对应为 4,0,0,0,1,0,0,2,3,4,5 的整形数组。

说明：　获取板卡硬件信息；

返回：　成功时返回 0，否则返回-1；

4.20　int StopMotionDetection(int hChannelHandle);

参数：　int hChannelHandle　　　　　　　通道句柄

说明：　停止运动检测；

返回：　成功时返回 0，否则返回-1；

4.21　int StartVideoCapture(int hChannelHandle);

参数：　int hChannelHandle　　　　通道句柄

说明：　启动数据截取， SDK 向用户注册的流处理回调函数发送数据流，用户可以在流处理函数中处理数据；

返回：　成功时返回 0，否则返回-1；

4.22　int StopVideoCapture(int hChannelHandle);

参数：　int　hChannelHandle　　　　通道句柄

说明：　停止数据截取；

返回：　成功时返回 0，否则返回-1；

4.23 int SetIBPMode(int hChannelHandle, int KeyFrameIntervals, int BFrames, int PFrames, int FrameRate);

参数：　int hChannelHandle,　　　　通道句柄
int KeyFrameIntervals,　　　　关键帧间隔（默认值为 25）
int BFrames,　　　　B 帧数（默认值为 2）
int PFrames,　　　　P 帧数
int FrameRate　　　　帧率（默认值为 25）

说明：　设置帧结构、关键帧间隔、B 帧数目、帧率，其中关键帧间隔不能小于 12，B 帧数目可以为 0、1 或 2，P 帧暂时设为无效，帧率范围 1-25，可在运行时设定；

注：　关键帧间隔解释：关键帧为编码码流中采用帧内压缩的图像帧，其特点是图像清晰度好，但数据量大，通常作为帧间编码的原始参考帧。关键帧间隔是连续的帧间编码的帧个数，因 MPEG-4 编码是有损压缩，关键帧的个数会影响图像质量，因此关键帧的间隔需要合理设计。

返回：　成功时返回 0，否则返回-1；

4.24 int SetDefaultQuant(int hChannelHandle, int IQuantVal, int PQuantVal, int BQuantVal);

参数：　int hChannelHandle,　　　　通道句柄
int IQuantVal,　　　　I 帧量化系数
int PQuantVal,　　　　P 帧量化系数
int BQuantVal　　　　B 帧量化系数

说明：　设置图像量化系数，用于调整图像质量，系数越低质量越好，取值范围（12-30），一般取值法为：取 I 帧和 P 帧一样大，B 帧比它们大 3 到 5，例如：15，15，20 和 18，18，23，这里系统默认值为 17，17，20；
若要设定高清晰度图像，推荐值为：12 12 17 并可在运行时设定。

注：量化系数解释：量化系数是强烈影响 MPEG 标准中编码图像质量和码率的参数，当量化系数越低，图像质量就会越高，码率也就越高，反之，图形质量就会越低，码率也就越低。

返回：　成功时返回 0，否则返回-1；

4.25　int SetOsd(int hChannelHandle, short Enable);

参数　int hChannelHandle,　　　　通道句柄
short Enable　　　　使能(0/1)

说明：　设置 OSD 显示，在当前的系统时间年月日星期时分秒叠加在预览视频之上，

并作透明处理，可以调整垂直方向的位置；

返回：　成功时返回 0，否则返回-1；

4.26　int SetAudioPreview(int hChannelHandle, short bEnable);

参数：　int hChannelHandle,　　通道句柄

short bEnable　　使能(0/1)

说明：　设置音频预览，打开或关闭，系统只能有一路打开；

返回：　成功时返回 0，否则返回-1;

4.27　int SetLogo(int hChannelHandle, int x, int y, int w, int h, char *yuv);

参数：　int hChannelHandle,　　通道句柄

int x,　　左上角位置 x(0-351)

int y,　　左 上 角 位 置　y(0-287(PAL)　0-239 (NTSC))

int w,　　宽度(0-128) (注：此宽度必须与原始 Logo 图像的宽度相一致)

int h,　　高度(0-64)

char *yuv　　YUV422 格式的图像指针

说明：　设置 OSD 屏幕图像位置及数据；用户程序可调用 LoadYUVFromBmpFile 从 24 位 bmp 文件中获取 YUV 数据（见 4.30）,透明处理由 DSP 完成。

返回：成功时返回 0，否则返回-1；

4.28　int StopLogo(int hChannelHandle);

参数：　int hChannelHandle　　通道句柄

说明：　停止 LOGO 显示；

返回：成功时返回 0，否则返回-1；

4.29 int LoadYUVFromBmpFile(char *FileName, unsigned char *yuv, int BufLen, int *Width, int *Height);

参数：　char *FileName　　文件名

unsigned char *yuv　　YUV422 格式的图像指针

int BufLen　　yuv 缓存大小

int *Width　　返回 yuv 图像的宽度

int *Height　　返回 yuv 图像的高度

说明：把 24 位 bmp 文件转成 yuv 格式的数据；

返回：成功时返回 0，否则返回-1；

4.30　int SaveYUVToBmpFile(char *FileName, unsigned char *yuv, int Width, int Height);

参数：char *FileName　　文件名

unsigned char *yuv　　YUV422 格式的图像指针

int Width　　yuv 图像的宽度

int Height yuv 图像的高度

说明：把 yuv 图像转成 bmp 文件；

4.31 int GetOriginalImage(int hChannelHandle, char *ImageBuf, int *Size);

参数：int hChannelHandle 通道句柄

char *ImageBuf 原始图像指针

int *Size 原始图像的大小（注：调用前是 imagebuf 的大小，调用后是实际图像所使用的字节数）

说明：获得原始图像，原始图像是标准的 CIF 图像格式(包括 QCIF 编码)，用户程序可调用 SaveYUVToBmpFile 来生成 24 位的 bmp 文件。

返回：成功时返回 0，否则返回-1；

4.32 int GetVideoSignal(HANDLE hChannelHandle);

参数：HANDLE hChannelHandle 通道句柄

说明：获得接入视频信号情况，用于检测视频丢失报警。

返回：返回 1 时表明视频信号正常，否则为 0；

4.33 int MotionAnalyzer(int hChannelHandle, char *MotionData, int iThreshold, int *iResult);

参数：int hChannelHandle 通道句柄

char *MotionData 运动矢量指针

int iThreshold 判断运动的一个区域阀值（0-100）

|int *iResult 按照区域阀值指定的强度分析后的结果，是一个数组，大小在 SetupMotionDetection 的 numberOfAreas 指定， 如果某数组单元的值大于零则表明有该区域有该值表明的运动强度。

说明：动态监测分析，运动检测由 DSP 完成，DSP 送出的 IPktMotionData 帧就是已经分析好的运动信息，，区域的运动分析由主机完成，数据源由码流中的 PktMotionData 帧提供（见 Demo 源码的 OndataReady 部分），结果在 iResult 中说明，1.2 版的运动分析基于 DSP 提供的运动强度，不再使用运动矢量，其灵敏度及可靠性有大幅提高，用户软件可使用由码流提供的运动强度信息来自己分析或调用该函数来进行区域分析，运动强度数据结构在 3.4 和 3.5 节说明；

返回：成功时返回 0，否则返回-1；

4.34 int AdjustMotionDetectPrecision(int hChannelHandle, int iGrade, int iFastMotionDetectFps, int iSlowMotionDetectFps)

参数：int hChannelHandle 通道句柄

iGrade 运动分析灵敏度等级（0-6)，（默认值为 2）

int iFastMotionDetectFps, 高速运动检测的帧间隔，取值范围 0-12，0 表示不作高速运动检测，通常值为 2，

（默认值为 2）

int iSlowMotionDetectFps 低速运动检测，适合于慢速运动情况，通常取值范围 13 以上，当取值为 0 时不作低速运动检测。（默认值为 200）

说明：调整运动分析灵敏度（iGrade），可于编码时动态调整运动侦测的灵敏度，决定 DSP 全局运动分析的灵敏度，与 MotionAnalyzer 的 iThreshold 不同，后者主要用于主机分析指定区域的运动统计结果，等级 0 最灵敏，6 最迟钝；推荐值为 2;

返回：成功时返回 0，否则返回-1；

4.35 int CaptureIFrame(int hChannelHandle)

参数：int hChannelHandle 通道句柄

说明：将目前编码帧强制设定为 I 帧模式，可从码流中提取该帧单独用于网络传送;

返回：成功时返回 0，否则返回-1；

**(2.2 版提供的新函数)**

4.36 int SetEncodePictureFormat(int hChannelHandle, PictureFormat_t PicFormat);

参数：int hChannelHandle 通道句柄

PictureFormat_t PicFormat 图像类型（CIF 或 QCIF 或 2CIF）

说明：设置当前通道的编码格式，必须在录像停止时调用,不使用本函数时，系统默认图像类型为 CIF；**3.2 版新增加了 2CIF 类型**

注意： 3.4 版本新增 CIFQCIF 和 CIFQQCIF 两种编码格式，

当某一通道被设置为 CIFQCIF 的模式，启动录像后，会分别送两种数据流：CIF 的和 QCIF 的，

3.8 版本新增 DCIF 编码格式

应用程序应通过函数 GetSubChannelStreamType()甄别处理。原来的编码格式不变。(QQCIF 的分辨率为 96*80)

返回：成功时返回 0，否则返回-1；

4.37 SetupBitrateControl(int hChannelHandle, int MaxBps)

参数：int hChannelHandle 通道句柄

int MaxBps 最大波特率（10000 以上）

说明：可用来设置最大波特率，MaxBps 设为 0 时码率控制无效，当设置为某一最大波特率时，当编码码流将超过该值时，DSP 会自动调整编码参数来保持不超过最大波特率，当码流低于最大波特率时，DSP 不进行干涉，调整误差为<10%。

4.38 SetLogoDisplayMode(int hChannelHandle, unsigned short ColorKeyR, unsigned short ColorKeyG, unsigned short ColorKeyB, unsigned short bTranslucent, int TwinkleInterval)

参数：int hChannelHandle 通道句柄

int ColorKeyR, ColorKeyG,, ColorKeyB, LOGO 图像该颜色将在显示时完全透明；

unsigned short bTranslucent LOGO 图像是否做半透明处理；

| | | |
|---|---|---|
| int | TwinkleInterval | 闪烁的时间设置，由 16 进制数表示为 0xXXYY,其中 XX 为显示的秒数，YY 为停止显示的秒数，XXYY 均为 0 时正常显示； |

说明：可通过该函数来设置 LOGO 的显示模式。

4.39 SetOsdDisplayMode(int hChannelHandle, int Brightness, unsigned short bTranslucent, int TwinkleInterval, unsigned short * Format1,unsigned short * Format2)

| | | |
|---|---|---|
| 参数：int | hChannelHandle | 通道句柄 |
| int | Brightness | OSD 显示亮度，255 最亮，0 最暗； |
| unsigned short | bTranslucent | OSD 图像是否做半透明处理； |
| Int | TwinkleInterval | V3.8 改动为：当值为1 时，OSD 的亮度根据背景的亮度来调整，但背景较亮时，OSD 亮度自动调低，但背景较暗时，OSD 亮度自动调亮。原来的闪烁功能关闭 |
| char | *Forma1, Format2 | 描述字符叠加的位置和次序的格式串，具体定义如下： |

USHORT X， USHORT Y， CHAR0， CHAR1， CHAR2，… CHARN， NULL

其中 X，Y 是该字串在标准 CIF 图像的起始位置，X 必须是 8 的倍数，Y 可以在图象高度内取值即（0-287）PAL 、（0-239）NTSC；CHARN 也是 USHORT 型的参数，可以是 ASCII 码也可以是汉字，当想要显示当前时间时，可以指定为固定的时间定义值，其值如下：

| | |
|---|---|
| _OSD_YEAR4 | 四位的年显示，如 2002 |
| _OSD_YEAR2 | 两位的年显示，如 02 |
| _OSD_MONTH3 | 英文的月显示，如 Jan |
| _OSD_MONTH2 | 两位阿拉伯数字的月显示，如 07 |
| _OSD_DAY | 两位的阿拉伯数字的日显示，如 31 |
| _OSD_WEEK3 | 英文的星期显示，如 Tue |
| _OSD_CWEEK1 | 中文的星期显示，如星期二 |
| _OSD_HOUR24 | 24 小时的时钟显示，如 18 |
| _OSD_HOUR12 | 12 小时的时钟显示，如 AM09 或 PM09 |
| _OSD_MINUTE | 两位分钟的显示 |
| _OSD_SECOND | 两位秒的显示 |

在格式字符串的最后必须以 NULL（0）结尾，否则会显示错误的内容。

字符串和时间显示可以在 FORMAT1 也可以在 FORAMT2，也可以混合在一起，但不得超过一行 CIF 图像的宽度。

要显示位置在 16，19 的字符串“办公室”的格式字符串如下：

USHORT Format[] = {16, 19, ‘办’,’公’,’室’, ‘\0’};

要显示位置在 8， 3 的时间字符串可以如下：

USHORT Format[]={8, 3, _*OSD*_YEAR4, ‘年’,_OSD_MONTH2,’月’,_OSD_DAY, ‘日’,_OSD_HOUR24,’:’, _*OSD*_MINUTE, ‘:’, _OSD_SECOND, ‘\0’};

如果只想显示其中一行，则将起始的字符串定义如下：

USHORT FormatNoDisplay[]={0, 0, '\0'};

## 版本 2.4 新增功能说明：

4.40 SetVideoStandard(int hChannelHandle, VideoStandard_t VideoStandard)

参数：int hChannelHandle 通道句柄

VideoStandard_t VideoStandard 视频标准

说明：设置视频标准，在某一制式的摄像头已经接好的情况下启动系统时可不必调用该函数，如果没有接摄像头的情况下启动系统然后再接 NTSC 制式的摄像头则必须调用该函数，或者中途调换不同制式的摄像头也必须调用该函数；

4.41 ResetDSP(int DspNumber)

参数：int DspNumber DSP 的索引号

说明：复位某 DSP 系统，注意请谨慎调用该函数时，请确定 DSP 故障无法软件恢复时再关闭相关的资源后复位 DSP；在调用该函数前无须调用 ChannelClose(),但之后要从新调用 ChannelOpen() 等一系列函数以恢复正常使用!

4.42 GetSoundLevel(int hChannelHandle)

参数：int hChannelHandle 通道句柄

说明：获取当前通道的现场声音幅度，注意当无声音输入时因背景躁声的原因返回值并不为 0；

(注：其他未说明函数为专用卡用户使用， 海康威视提供其他文档说明。)

## 版本 3.0 新增功能说明：

```
typedef enum{
        brCBR = 0,
        brVBR = 1,
}BitrateControlType_t;
```

4.43 int SetBitrateControlMode(int hChannelHandle, BitrateControlType_t brc)

参数：int hChannelHandle 通道句柄

BitrateControlType brc 码流类型

说明：设置码流类型，可设为定码流和变码流模式,定码流的码流值可用 SetupBitrateControl（）来设定。

## 版本 3.4 新增功能说明：

4.44 int SetupSubChannel(int hChannelHandle, int iSubChannel);

参数：int hChannelHandle 通道句柄

int iSubChannel 子通道号

说明：配合 CIFQCIF 和 CIFQQCIF 模式使用，当设置 CIFQCIF(CIFQQCIF)模式时，子

通道 0 编码 CIF，子通道 1 编码 QCIF(QQCIF)，这时可对子通道 0，1 分别设置某些参数。帧率、关键帧间隔、OSD、LOGO、STREAMTYPE 等参数对子通道 0、1 是一样的；在设置量化系数、变码流/定码流模式、码流大小等参数时应调用此函数分别对子通道 0、1 进行设置，缺省是对子通道 0 进行设置

4.45 Int GetSubChannelStreamType(void *DataBuf, int FrameType);

参数：void *DataBuf 输入数据缓存区

int FrameType 输入帧类型

说明：配合 CIFQCIF 和 CIFQQCIF 模式使用，当设置 CIFQCIF(CIFQQCIF)模式时，子通道 0 编码 CIF，子通道 1 编码 QCIF(QQCIF)，启动录像后，会送两种数据的混合流，CIF 和 QCIF(QQCIF)，调用此函数得到数据包类型，供应用程序使用以进行对数据的份流。

返回： 0 其他数据

1 CIF 数据流的文件头

2 QCIF/QQCIF 数据流的文件头

3 CIF 数据流的视频帧类型

4 QCIF/QQCIF 数据流的视频帧类型

5 数据流的音频帧

## 版本 3.8 新增函数说明：

4．46 int SetupMask(int hChannelHandle, RECT *rectList, int iAreas)

参数： int hChannelHandle 通道句柄

RECT * rectList 矩形框数组

int iAreas, 矩形框个数

说明：启动屏幕遮挡函数，最多可设置 3 2 个，矩形框的范围是(0,0,703,575)，矩形框的宽度为16 对齐，高度为8 对齐

4 .47 int StopMask(int hChannelHandle)

参数： int hChannelHandle 通道句柄

说明： 停止屏幕遮挡函数。

4.48 int StartVideoPreviewEx(int hChannelHandle, RECT PreviewRect, int* pAddr, int VideoFormat, int win_width)

参数： int hChannelHandle 通道句柄

RECT PreviewRet, 矩形显示区域

Int pAddr 在选用 YUV 格式时，会返回一个 YUV 数据地址，用户可从该地址获得原始的预览数据，格式为 UYVY, pitch 值为 2 倍的所设矩形显示区的宽度，数据大小为 pitch * 显示区域的高；

int VideoFormat, 视频格式,现可选择 YUV, RGB16, RGB24

int win_width 视窗的分辨率宽度，注意：在 x-window 为 800*600 时，实际宽度为 832。

说明： 启动视频预览，

返回： 成功时返回 0，否则返回-1；

# 5．补充说明

5． 1 用户在 OpenChannel 中注册的回调函数必须定义为如下形式：

void＊StreamReadCallBack(int ChannelNum,void＊DataBuf, int FrameType ,int Length,int FrameNum);

| | |
|---|---|
| int ChannelNum | 通道号 |
| void * DataBuf | 帧数据区 |
| int FrameType | 帧类型 |
| int Length | 帧数据长度 |
| int FrameNum | 帧顺序号 |

用户主要在这个回调函数中处理数据，包括录象，运动检测等。。。

在这个回调函数中，可能收到的帧类型有

PktSysHeader 系统头

每次 StartVideoCapture()之后,DSP 会送上来一个系统头，每个录象文件必须以系统头开始，否则无法回放。

PktIFrames I 帧包

PktPFrames P 帧包

PktBBPFrames BBP 帧包

PktAudioFrames 音频帧包

以上 5 种帧都是有效的数据类型，都应该写入录象文件。

PktMotionDetection 动态监测包

在 StartMotionDetection()之后，该类帧会出现

PktSFrames

运行 CaptureIFrame()后，DSP 送上来的 IFrame 数据。

# 6． yuv 格式文件与 bmp 格式文件之间的转换

## 6.1 yuv 格式文件转到 bmp 格式文件

yuv2bmp <yuv 格式文件> <bmp 格式文件> <width> <height>

其中：width height 必须和 YUV 图象的宽和高保持一致，由于 DSP 送上来的 YUV 图象的宽和高固定为 352*288，所以

在这里，width=352 height=288

例如：yuv2bmp /home/demo/c001.yuv /home/demo/c001.bmp 352 288

再进到 yuv 格式文件所在目录下可以看到有一个新的 bmp 格式文件生成。

## 6.2 bmp 格式文件转到 yuv 格式文件

bmp2yuv <bmp 格式文件> <yuv 格式文件> <width> <height>

其中：width height 为所需输出 YUV 图象的宽和高。

再进到 bmp 格式文件所在目录下可以看到有三个新的文件生成，分别为.y 后缀，.u 后缀和.v 后缀。

如有任何问题，请和我们联系：
Tel: 0571-88075998-8029
Mail: yingxq@hikvision.com